

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平5-89303

(43)公開日 平成5年(1993)4月9日

(51)Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 K 19/07				
G 0 6 F 3/08		C 7165-5B		
G 0 6 K 17/00		D 8623-5L		
G 1 1 C 7/00	3 1 5	7323-5L		
		8623-5L		
			G 0 6 K 19/ 00	N
			審査請求 未請求	請求項の数1(全 15 頁)

(21)出願番号 特願平3-21080

(22)出願日 平成3年(1991)2月14日

(71)出願人 000003193

凸版印刷株式会社

東京都台東区台東1丁目5番1号

(72)発明者 平野 誠治

東京都台東区台東一丁目5番1号 凸版印刷株式会社内

(72)発明者 高橋 正志

東京都台東区台東一丁目5番1号 凸版印刷株式会社内

(72)発明者 寄本 義一

東京都台東区台東一丁目5番1号 凸版印刷株式会社内

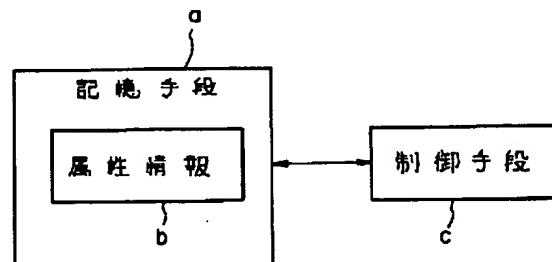
(74)代理人 弁理士 志賀 正武 (外2名)

(54)【発明の名称】 ICカード

(57)【要約】

【目的】 アドレス方式が異なるアプリケーションに対しても容易に対応することができるICカードを実現する。

【構成】 ICカードの記憶手段を少なくとも1つ以上に分割し、この分割された記憶領域毎にアドレス空間を定義する。そして、このアドレス空間毎に各種アドレス方式を識別するための属性情報を設ける。アプリケーション側がこの記憶手段にアクセスする際には、ICカードに搭載される制御手段が前記属性情報に対応したアドレス方式に従って入出力制御を行う。



【特許請求の範囲】

【請求項1】 少なくとも1つ以上に分割された記憶領域毎に定義されたアドレス空間を有し、前記アドレス空間毎にアドレス方式を識別する属性情報を設けた記憶手段と、

前記属性情報に対応したアドレス方式で入出力制御する制御手段とを具備することを特徴とするICカード。

【発明の詳細な説明】

【産業上の利用分野】この発明は、記憶領域でのアドレス方式が異なるアプリケーションを利用可能とするICカードに関する。

【0001】

【従来の技術】周知のように、近年、いわゆるICカードと総称するカードが多方面に採用されており、例えばバンクカード、IDカード、クレジットカード等にCPU、メモリを内蔵したICカードや、電子手帳または携帯型のパーソナルコンピュータ等の外部記憶装置として単にデータを記憶するメモリだけを搭載したICメモリカードが使われている。

【0002】この種のカードに搭載される記憶手段としては、ROM、EPROM等の不揮発性メモリ、或いはDRAM、SRAM等の揮発性メモリが用いられている。特に、CPU、メモリを内蔵したICカードでは、1チップ上に両者を組込形成したマイクロプロセッサが多用されている。このようなICカードは、カードリーダーライタを介してデータの授受を行うものであり、それぞれICカードのメモリへのアクセスは、内蔵のCPUにより行われ、メモリカードのメモリへのアクセスは、端末側のCPUにより行われるようになっている。

【0003】このアクセスの際には、所定のメモリアドレス方式に応じたデータの読み出しおよび書込みがなされる。このアドレス方式には、論理アドレス方式、物理アドレス方式およびレコード管理方式がある。まず、この論理アドレス方式とは、メモリを管理するソフトウェアにおいて定義された相対アドレスによって、当該メモリのアドレス空間を規定し、この相対アドレスに基づいて所定の記憶エリアにアクセスするようにした方式である。なお、この相対アドレスは、所定の変換テーブルにより実際のメモリアドレス（絶対アドレス）に変換される。物理アドレス方式とは、上記絶対アドレスでアドレス空間を規定し、この絶対アドレスに基づいて所定の記憶エリアにアクセスする方式である。レコード管理方式とは、メモリ上の記憶エリアを所定データ長毎のレコードに区分し、区分したレコードを識別するレコード番号またはレコードポインタを用いて記憶エリアを管理する方式である。故に、データの授受を行う場合には、適用されるカードのアドレス方式に準拠したアプリケーションを具備し、それに基づいてデータの読み出し、書込みとを行うように構成されている。

【0004】

【発明が解決しようとする課題】ところで、従来のICカードにおいては、上述した各種アドレス方式のいずれかによってメモリの記憶エリアが管理されており、これら各アドレス方式に対応するアプリケーションが種々混在した状況下にある。したがって、ICカードを異なるアドレス方式のアプリケーションに対応させるためには、これらアドレス方式を個々に管理するマイクロチップをカード内に具備しなければならないが、こうした場合、該カードのコスト上昇などの様々な不都合を招致する。このため、現実的には、ICカードのアドレス方式に対応したアプリケーションでなければ、カードのデータにアクセスすることができないという欠点があった。この発明は上述した事情に鑑みてなされたもので、アドレス方式が異なるアプリケーションに容易に対応することができるICカードを提供することを目的としている。

【0005】

【課題を解決するための手段】この発明は、図1に示すように、少なくとも1つ以上に分割された記憶領域毎に定義されたアドレス空間を有し、前記アドレス空間毎にアドレス方式を識別する属性情報bを設けた記憶手段aと、前記属性情報bに対応したアドレス方式で入出力制御する制御手段cとを具備することを特徴としている。

【0006】

【作用】上記構成によれば、記憶手段aには、少なくとも1つ以上に分割された記憶領域毎に定義されたアドレス空間を有し、このアドレス空間毎にアドレス方式を識別する属性情報bが設けられ、制御手段cが前記属性情報bに対応したアドレス方式に従って入出力制御を行う。これにより、アドレス方式が異なるアプリケーションに対応可能となる。

【0007】

【実施例】以下、図面を参照してこの発明の実施例について説明する。

A. 実施例の構成

図2は、この発明による一実施例の構成を示すブロック図である。この図に示すように、ICカード11は、カードリーダーライタ21との間でコマンド、データの送受信を行うI/O12と、これらのコマンド処理、あるいはデータ処理のプログラムやデータなどが格納されるEEPROM13と、データ通信の制御およびこれらのプログラムに基づいてのデータ処理を行うCPU14と、このCPU14のワークエリアとして用いられるRAM15とを具備してなる。

【0008】また、このICカード11との間でデータの授受を行うカードリーダーライタ21は、I/O21と、ROM23と、CPU24と、RAM25とから構成されている。なお、このカードリーダーライタ21は、通信回線を介し、ホストコンピュータ側とデータ通信できるようにしている。

【0009】このICカード11において、EEPROM

M13の記憶エリアは、システム情報を格納しておくシステム領域と、ユーザ情報を格納しておくユーザ領域とに分れており、さらに、図3に示すファイル構造となっている。図3に示すように、このICカード11は、システムディレクトリSDの下に3つのルートディレクトリRT1~RT3を構成しており、これらによってファイル管理がなされている。ここで、ディレクトリRT1は前述した物理アドレス方式のファイルが管理される領域、ディレクトリRT2は前述した論理アドレス方式のファイルが管理される領域、また、ディレクトリRT3は前述したレコード管理方式のファイルが管理される領域である。このように、この実施例によるICカードは、メモリの所定エリアが3分割され、それぞれ各アドレス方式により管理されるように構成されている。なお、このICカードには、CPUが具備され、これにより各アドレス方式のファイル管理がなされるように構成されている。

【0010】これらディレクトリRT1~RT3は、それぞれ図4に示すフォーマットでアドレス管理情報が登録されている。すなわち、図4(イ)は物理アドレス方式の管理フォーマット、図4(ロ)は論理アドレス方式の管理フォーマットを示し、図4(ハ)はレコード管理方式の管理フォーマットを示している。この図において、F1~Fnは、上記の各フォーマットを形成する所定バイト長のフレームである。例えば、F1はファイル名が登録される8バイト長のフレーム、F2は上述したアドレス方式を識別するための格納形式が登録される1バイト長のフレームである。また、F3はフレームF1に定義されたファイル名の先頭アドレスが登録されるフレームである。なお、Fnはブロックチェックコードであり、このアドレス管理情報をチェックする際に用いられる。

【0011】図5は、上記フレームF2の記述形式を示す図である。この図に示すように、1バイト中の下位3ビットでアドレス方式を識別するようになっている。このフレームF2において、例えば、下位3ビットが「001」の場合には物理アドレス方式、「010」の場合には論理アドレス方式、「100」の場合にはレコード管理方式を各々表わす。そして、この下位3ビットが物理アドレス方式を表わしている時には、図4(イ)のフレームF4、F5、F6はいずれも未使用となる。一方、下位3ビットが論理アドレス方式を表わしている場合、図4(ロ)のフレームF5には、アドレスポインタが登録されると共に、フレームF6にエリアサイズが登録される。このアドレスポインタとは、論理アドレスを絶対(物理)アドレスに変換する際に用いられる変換テーブルの入力番地を指し示すものである。さらに、下位3ビットがレコード管理方式を表わしている場合、図4(ハ)のフレームF4には、格納レコード数が登録され、かつ、フレームF5に次レコード番号、フレームF

6にレコードサイズがそれぞれ登録される。

【0012】B. 実施例の動作

次に、上記構成によるICカード11にアクセスする場合の動作について図6~図20を参照し、説明する。なお、この動作説明においては、ICカード11がカードリーダーライタ21にセットされ、該カードリーダーライタ21により所定のファイルがアクセスされるものとする。まず、カードリーダーライタ21にICカード11がセットされると、該カード内部に搭載されたCPU14が起動する。これにより、図6に示すメインルーチンが開始し、ステップSa1に進む。ステップSa1では、イニシャライズが行われ、この結果、各種内部レジスタなどがリセットされた後、カードリーダーライタ21側に対してATR情報(リセットした旨を表わす情報)が出力される。次に、ステップSa2に進むと、CPUはカードリーダーライタ側からコマンドを受けるまで待機状態となる。ここで、カードリーダーライタ側から後述する各種コマンドを受けると、ステップSa2の判断結果が「YES」となり、次のステップSa3に進む。ステップSa3では、受けたコマンドに応じたコマンド処理ルーチンが実行される。そして、ステップSa4では、このコマンド処理ルーチンの実行結果を所定のフォーマットに基づくレスポンスデータとしてカードリーダーライタ側へ送出する。そして、以後、上述したステップSa2~Sa4を繰り返す動作となる。

【0013】以下では、このようなステップSa2~Sa4の動作において、前述した各アドレス方式のファイルにアクセスする場合について詳述する。

①所定のファイルを開く場合の動作

この場合、カードリーダーライタ21側からICカード11へオープンコマンドが供給される。なお、このオープンコマンドのコマンドフォーマットは図8(イ)に示す通りである。このようなオープンコマンドが供給されると、前述したメインルーチンにおけるステップSa2の判断結果が「YES」となり、ステップSa3に進む。これにより、図7に示すオープンコマンド処理ルーチンが起動され、ステップSb1に進む。ステップSb1では、ステータスデータSTS1を「0」とし、次のステップSb2へ進む。ここで、ステータスデータSTS1とは、コマンド処理ルーチンにおけるエラーの有無を表わすデータである。ステップSb2では、上記オープンコマンドに付与されるファイル名FNに基づいて、前述のルートディレクトリRT1~RT3をサーチする。そして、ステップSb3では、サーチした結果、ファイル名FNに該当するファイルが存在しているか否かを判断する。ここで、ファイル名FNに該当するファイルが存在する場合には、判断結果が「YES」となり、次のステップSb4へ進む。ステップSb4では、ファイルが存在するディレクトリのフレームF3にファイルの先頭アドレスをセットする。次いで、ステップSb5では、

このディレクトリのアドレス管理情報を読み出し、このルーチンを終了する。

【0014】一方、上述したステップSb3の判断結果が「NO」の場合、すなわち、オープンコマンドに付与されたファイル名FNに該当するファイルが存在しない場合には、ステップSb6に進む。ステップSb6では、ステイタスデータSTS1に「4」を加算し、このルーチンを終了する。そして、CPU14の処理は、このオープンコマンド処理ルーチンからメインルーチンに戻り、レスポンス送信処理（ステップSa4）が行われ、この場合のレスポンス送信処理では、図8（ロ）に示すフォーマットのレスポンスデータがICカード11からカードリーダーライタ21側へ送出される。ここで、ステイタスデータSTS1が「4」であると、カードリーダーライタ21側は該当するファイルが存在しない旨のエラー表示を行う。

【0015】②物理アドレス方式のファイルにアクセスする場合の動作

上述したオープンコマンド処理によって開かれたファイルが物理アドレス方式で管理されている場合には、以下に示す読み出し動作および書き込み動作が行われる。

a. 読み出し動作

この読み出し動作では、カードリーダーライタ21側からICカード11へ図10（イ）に示すフォーマットの読み出しコマンドが供給される。このような読み出しコマンドが供給されると、図9に示す読み出しコマンド処理ルーチンが起動され、CPU14の処理がステップSc1に進む。ステップSc1では、ステイタスデータSTS1を「0」とし、次のステップSc2へ進む。ステップSc2では、ディレクトリRT1のフレームF2が物理アドレスを表わす格納形式であるか否かを判断する。そして、フレームF2の下位3ビットが「001」である場合には、この判断結果が「YES」となり、ステップSc3に進む。ステップSc3では、読み出しコマンドに付与されたアドレスRAを読み取り、これを内部レジスタにセットする。次に、ステップSc4では、このアドレスRAから順次ファイルのデータを読み出す。この読み出しの際には、読み出しコマンドに含まれるデータ長LEN（図10（イ）参照）を読み出し単位としている。次いで、ステップSc5では、データ長LEN毎の読み出しにおいて、アドレスエラーがあるか否かを判断する。そして、エラーが無い場合には、ステップSc6に進み、読み出したデータを図示していない出力バッファにセットし、このルーチンを終了する。

【0016】一方、上述したステップSc2の判断結果が「NO」の場合、すなわち、このファイルが物理アドレス方式でない場合には、ステップSc7に進み、ステイタスデータSTS1に「1」を加算する。また、読み出し時にアドレスエラーがあった場合には、ステップSc5の判断結果が「NO」となり、ステップSc8に進

み、このステイタスデータSTS1に更に「1」を加算する。そして、CPU14の処理がメインルーチンのレスポンス送信処理（ステップSa4）に移行すると、図10（ロ）に示すフォーマットによるレスポンスデータがICカード11側からカードリーダーライタ21側へ送出される。この結果、読み出しデータがカードリーダーライタ21側に受取られることになる。なお、このレスポンスデータにおけるステイタスデータSTS1が「2」の場合、カードリーダーライタ21側は、上述した読み出しコマンド処理にエラーがあったとしてエラー表示を行う。

【0017】b. 書き込み動作

この書き込み動作では、カードリーダーライタ21側からICカード11へ図12（イ）に示すフォーマットの書き込みコマンドが供給される。このような書き込みコマンドが供給されると、図11に示す書き込みコマンド処理ルーチンが起動され、ステップSd1に進む。ステップSd1では、ステイタスデータSTS1を「0」とし、次のステップSd2へ進む。ステップSd2では、ディレクトリRT1のフレームF2が物理アドレスを表わす格納形式であるか否かを判断する。そして、フレームF2の下位3ビットが「001」である場合には、この判断結果が「YES」となり、ステップSd3に進む。ステップSd3では、書き込みコマンドに付与された書き込みアドレスWAを読み取り、これを内部レジスタにセットする。次に、ステップSd4では、オープンしたファイルの書き込み領域が不足しているか否かを判断する。そして、書き込み領域が不足していない場合には、ここでの判断結果が「NO」となり、次のステップSd5に進む。ステップSd5では、書き込みアドレスWAから順次、書き込みコマンドに付与された書き込みデータWDの書き込みが行われ、このルーチンを終了する。

【0018】一方、上述したステップSd2の判断結果が「NO」の場合、すなわち、このファイルが物理アドレス方式でない場合には、ステップSd6に進み、ステイタスデータSTS1に「2」を加算する。また、ファイルの書き込み領域が不足している場合には、ステップSd4の判断結果が「YES」となり、ステップSd7に進み、このステイタスデータSTS1に更に「1」を加算する。そして、CPUの処理がメインルーチンのレスポンス送信処理（ステップSa4）に移行すると、図12（ロ）に示すフォーマットのレスポンスデータがICカード11からカードリーダーライタ21側へ送出される。ここで、このレスポンスデータのステイタスデータSTS1が「3」の場合、カードリーダーライタ21側は上述した書き込みコマンド処理にエラーがあったとしてエラー表示を行う。

【0019】③論理アドレス方式のファイルにアクセスする場合の動作

前述したオープンコマンド処理によって開かれたファイ

ルが論理アドレス方式で管理されている場合には、以下に示す読み出し動作および書き込み動作が行われる。

a. 読み出し動作

この読み出し動作では、カードリーダライタ側からICカードへ図15(イ)に示すフォーマットの読み出しコマンドが供給される。このような読み出しコマンドが供給されると、図13に示す読み出しコマンド処理ルーチンが起動され、CPU14の処理がステップSe1に進む。ステップSe1では、ステータスデータSTS1を「0」とし、次のステップSe2へ進む。ステップSe2では、ディレクトリRT2のフレームF2が論理アドレスを表わす格納形式で記述されているか否かを判断する。そして、このフレームF2の下位3ビットが「010」である場合には、この判断結果が「YES」となり、ステップSe3に進む。ステップSe3では、オープンしたファイルの先頭アドレスをセットする。次に、ステップSe4では、フレームF5に登録されるアドレスポインタの値を内部レジスタにセットする。

【0020】次いで、ステップSe5では、セットされた先頭アドレスを前記アドレスポインタに応じて絶対アドレスに変換し、この絶対アドレスから順次、ファイルのデータを読み出す。この読み出しの際には、読み出しコマンドに含まれるデータ長LEN(図15(イ)参照)を読み出し単位としている。次に、ステップSe6では、読み出し時にファイル終端を表わすEOFコードを検出したか否かが判断される。ここで、このEOFコードが検出されない場合には、ファイル終端ではないので、EOFエラーとはならず、この判断結果は「NO」となり、次のステップSe7に進む。そして、ステップSe7では、読み出したデータを図示していない出力バッファにセットする。続いて、ステップSe8では、読み出し完了後の論理アドレスを示すように、前述したフレームF3のアドレスポインタを更新し、このルーチンを終了する。

【0021】一方、上述したステップSe2の判断結果が「NO」の場合、すなわち、このファイルが論理アドレス方式でない場合には、ステップSe9に進み、ステータスデータSTS1に「1」を加算する。また、読み出し時にファイル終端となってしまう、EOFエラーが発生した場合には、上述したステップSe6の判断結果が「YES」となってステップSe10に進み、このステータスデータSTS1に更に「1」を加算する。そして、CPU14の処理が前述したメインルーチンのレスポンス送信処理(ステップSa4)に移行すると、図15(ロ)に示すフォーマットによるレスポンスデータがICカード11からカードリーダライタ21側へ送出される。これにより、読み出しデータがカードリーダライタ21側に受取られることになる。一方、このレスポンスデータのステータスデータSTS1が「2」の場合、カードリーダライタ21側は、上述した読み出しコマ

ンド処理にエラーがあったとしてエラー表示を行う。

【0022】b. 書き込み動作

この書き込み動作では、カードリーダライタ側からICカードへ図16(イ)に示すフォーマットの書き込みコマンドが供給される。このような書き込みコマンドが供給されると、図14に示す書き込みコマンド処理ルーチンが起動され、ステップSf1に進む。ステップSf1では、ステータスデータSTS1を「0」とし、次のステップSf2へ進む。ステップSf2では、ディレクトリRT2のフレームF2が論理アドレスを表わす格納形式であるか否かを判断する。そして、フレームF2の下位3ビットが「010」である場合には、この判断結果が「YES」となり、ステップSf3に進む。ステップSf3では、オープンしたファイルの先頭アドレスをセットし、続いて、ステップSf4では、ディレクトリRT2のフレームF5に登録されるアドレスポインタの値を参照し、この値を内部レジスタにセットする。

【0023】次に、ステップSf5では、ファイルの書き込み領域が不足しているか否かを判断する。そして、書き込み領域が不足していない場合には、ここでの判断結果が「NO」になり、次のステップSf6に進む。ステップSf6では、セットされた先頭アドレスを前記アドレスポインタに応じて絶対アドレスに変換し、この絶対アドレスから順次、書き込みコマンドに付与された書き込みデータWDの書き込みを行う。次いで、ステップSf7では、書き込み完了後の論理アドレスを示すように、前述したフレームF3のアドレスポインタを更新し、このルーチンを終了する。一方、上述したステップSf2の判断結果が「NO」の場合、すなわち、このファイルが論理アドレス方式でない場合には、ステップSf8に進み、ステータスデータSTS1に「2」を加算する。また、ファイルの書き込み領域が不足している場合には、ステップSf5の判断結果が「YES」となってステップSf9に進み、このステータスデータSTS1に更に「1」を加算する。

【0024】そして、CPU14の処理が前述したメインルーチンのレスポンス送信処理(ステップSa4)に移行すると、図16(ロ)に示すフォーマットのレスポンスデータがICカード11側からカードリーダライタ21側へ送出される。ここで、このレスポンスデータのステータスデータSTS1が「3」の場合、カードリーダライタ21側は、上述した書き込みコマンド処理にエラーがあったとしてエラー表示を行う。

【0025】④レコード管理方式のファイルにアクセスする場合の動作

前述したオープンコマンド処理によって開かれたファイルがレコード管理方式で管理されている場合には、以下に示す読み出し動作および書き込み動作が行われる。

a. 読み出し動作

この読み出し動作では、カードリーダライタ側からIC

カードへ図19(イ)に示すフォーマットの読み出しコマンドが供給される。なお、このフォーマットにおけるレコード番号Nは、読み出すレコードの番号を指定するものである。このレコード番号Nが「0」である時は、次のレコード番号を指し、通常は、読み出しするレコード番号を1~255の範囲で表わす。このような読み出しコマンドが供給されると、図17に示す読み出しコマンド処理ルーチンが起動され、ステップSg1に進む。ステップSg1では、ステータスデータSTS1を「0」とし、次のステップSg2へ進む。ステップSg2では、ディレクトリRT3のフレームF2がレコード管理を表わす格納形式で記述されているか否かを判断する。そして、このフレームF2の下位3ビットが「100」である場合には、この判断結果が「YES」となり、ステップSg3に進む。ステップSg3では、オープンしたファイルにおけるレコードの先頭アドレスをフレームF3から読み出す。そして、ステップSg4では、上記レコード番号Nが「0」であるか否かを判断する。ここで、レコード番号Nが「0」でない場合には、判断結果が「NO」になり、次のステップSg6に進む。一方、レコード番号Nが「0」であると、この判断結果が「YES」となり、ステップSg5に進む。ステップSg5では、フレームF5に登録された次レコード番号を読み取り、これを内部レジスタにセットする。【0026】次に、ステップSg6では、フレームF4から読み出した格納レコード数が、読み出しを行うレコード番号N以上であるか否か、つまり、読み出しするレコード番号Nが妥当な値であるかどうかを判断する。ここで、このレコード番号が妥当な値である場合には、判断結果が「YES」となり、次のステップSg7に進む。ステップSg7では、フレームF6のレコードサイズを読み取り、このレコードサイズと、レコード番号Nおよび先頭アドレスとから、読み出しを開始すべき絶対アドレスを求める。すなわち、レコードサイズと次レコード番号(N-1)とを乗じ、この結果を先頭アドレスに加算する処理がなされる。そして、ステップSg8では、こうした処理により求められた絶対アドレスから順次、レコードのデータを読み出す。次いで、ステップSg9では、この読み出したデータを図示していない出力バッファにセットし、このルーチンを終了する。【0027】一方、上述したステップSg2の判断結果が「NO」の場合、すなわち、このファイルがレコード管理方式によらない場合には、ステップSg10に進み、ステータスデータSTS1に「2」を加算する。また、読み出しレコード番号が格納レコード数より小さく、不正なものである時には、上述したステップSg6の判断結果が「NO」となってステップSg11に進み、このステータスデータSTS1に更に「1」を加算される。そして、CPU14の処理が前述したメインルーチンのレスポンス送信処理(ステップSa4)に移行

すると、図19(ロ)に示すフォーマットによるレスポンスデータがICカード11からカードリーダーライタ21側へ送出される。これにより、読み出したデータがカードリーダーライタ21側に受取られることになる。ところで、このレスポンスデータのステータスデータSTS1が「3」となった場合、カードリーダーライタ21は、上述した読み出しコマンド処理にエラーがあったとしてエラー表示を行う。

【0028】b. 書込み動作

この書込み動作では、カードリーダーライタ側からICカードへ図20(イ)に示すフォーマットの書込みコマンドが供給される。なお、このフォーマットにおけるレコード番号Nは、書込みを行うレコード番号を指定するものである。このような書込みコマンドが供給されると、図18に示す書込みコマンド処理ルーチンが起動され、ステップSj1に進む。まず、ステップSj1では、ステータスデータSTS1を「0」とし、次のステップSj2へ進む。ステップSj2では、ディレクトリRT3のフレームF2がレコード管理を表わす格納形式で記述されているか否かを判断する。そして、このフレームF2の下位3ビットが「100」である場合には、この判断結果が「YES」となり、ステップSj3に進む。ステップSj3では、オープンしたファイルにおけるレコードの先頭アドレスをフレームF3から読み出し、これを内部レジスタにセットする。次いで、ステップSj4では、書込みコマンドに付与されたレコード番号Nが「0」であるか否かを判断する。ここで、レコード番号Nが「0」でない場合には、判断結果が「NO」になり、次のステップSg6に進む。一方、レコード番号Nが「0」であると、この判断結果が「YES」となり、ステップSj5に進む。ステップSj5では、フレームF5に登録されたレコード番号より1大きい次のレコード番号を内部レジスタにセットし、次のステップSj6に進む。

【0029】ステップSj6では、フレームF4から読み出した格納レコード数が、書込みを行うレコード番号N以上であるか否か、すなわち、書込みを行うレコード番号Nが妥当な値であるかどうかを判断する。ここで、このレコード番号が妥当な値である場合には、判断結果が「YES」となり、次のステップSj7に進む。ステップSj7では、フレームF6に登録されたレコードサイズを読み取り、このレコードサイズと、レコード番号Nおよび先頭アドレスとから、書込みを開始する絶対アドレスを求める。すなわち、レコードサイズと次レコード番号(N-1)とを乗じ、この結果を先頭アドレスに加算する処理がなされる。そして、ステップSj8では、こうした処理により求められた絶対アドレスから順次、書込みコマンドに付与された書込みデータWDを書込む。次いで、ステップSj9では、フレームF4に登録される格納レコード数を1インクリメントし、このル

ーチンを終了する。

【0030】ところで、上述したステップS j 6において、書き込みを行うレコード番号Nが不正な値であると、判断結果が「NO」となり、ステップS j 10に進む。ステップS j 10では、フレームF 6に登録されたレコードサイズに基づき、ファイル中にレコードエリアを確保する。次いで、ステップS j 11では、このレコードエリアを確保する際に、書き込み領域が不足しているか否かを判断する。ここで、書き込み領域が足りた場合には、この判断結果が「NO」となり、次のステップS j 12に進む。ステップS j 12では、書き込みを行うレコード番号として、現在のレコード番号を1インクリメントし、上記ステップS j 7に進む。これにより、新たなレコードに書き込みが行われることになる。

【0031】一方、上述したステップS j 2の判断結果が「NO」の場合、すなわち、このファイルがレコード管理方式によらない場合には、ステップS j 13に進み、ステータスデータSTS 1に「1」を加算する。また、レコードエリアを拡張する際に、書き込み領域が不足した時には、上記ステップS j 11の判断結果が「YES」となってステップS j 14に進み、このステータスデータSTS 1に更に「2」を加算する。そして、CPU 14の処理が前述したメインルーチンのレスポンス送信処理（ステップSa 4）に移行すると、図20（ロ）に示すフォーマットによるレスポンスデータがICカード11からカードリーダーライタ21側へ送出される。ところで、このレスポンスデータのステータスデータSTS 1が「3」となった場合、カードリーダーライタ21は、上述した読み出しコマンド処理にエラーがあったとしてエラー表示を行う。

【0032】このように、上述した実施例にあっては、各アドレス方式に対応させたファイルをICカードに具備させると共に、各アドレス方式に応じた読み出しコマンドおよび書き込みコマンドを実行することができるので、どのようなカードリーダーライタのアプリケーションにも対応可能になっている。

【0033】

【発明の効果】以上説明したように、この発明によれば、記憶手段には、少なくとも1つ以上に分割された記憶領域毎に定義されたアドレス空間を有し、このアドレス空間毎にアドレス方式を識別する属性情報が設けられ、制御手段が前記属性情報に対応したアドレス方式に従って入出力制御を行うので、アドレス方式が異なるアプリケーションに容易に対応することができる。

【図面の簡単な説明】

【図1】この発明の構成を示す機能ブロック図。

【図2】この発明の一実施例の構成を示すブロック図。

【図3】同実施例におけるファイル構造を示す図。

【図4】同実施例におけるディレクトリRT 1～RT 3のフォーマットを示す図。

【図5】同実施例におけるフレームF 2の格納形式の一例を示す図。

【図6】同実施例におけるメインルーチンを示すフローチャート。

【図7】同実施例におけるオープンコマンド処理ルーチンを示すフローチャート。

【図8】同実施例におけるオープンコマンド及び該コマンドに対するレスポンスデータのフォーマットを示す図。

【図9】同実施例における物理アドレス方式の読み出しコマンド処理ルーチンを示すフローチャート。

【図10】同実施例における物理アドレス方式時の読み出しコマンド及び該コマンドに対するレスポンスデータのフォーマットを示す図。

【図11】同実施例における物理アドレス方式の書き込みコマンド処理ルーチンを示すフローチャート。

【図12】同実施例における物理アドレス方式時の書き込みコマンド及び該コマンドに対するレスポンスデータのフォーマットを示す図。

【図13】同実施例における論理アドレス方式の読み出しコマンド処理ルーチンを示すフローチャート。

【図14】同実施例における論理アドレス方式の書き込みコマンド処理ルーチンを示すフローチャート。

【図15】同実施例における論理アドレス方式時の読み出しコマンド及び該コマンドに対するレスポンスデータのフォーマットを示す図。

【図16】同実施例における論理アドレス方式時の書き込みコマンド及び該コマンドに対するレスポンスデータのフォーマットを示す図。

【図17】同実施例におけるレコード管理方式の読み出しコマンド処理ルーチンを示すフローチャート。

【図18】同実施例におけるレコード管理方式の書き込みコマンド処理ルーチンを示すフローチャート。

【図19】同実施例におけるレコード管理方式時の読み出しコマンド及び該コマンドに対するレスポンスデータのフォーマットを示す図。

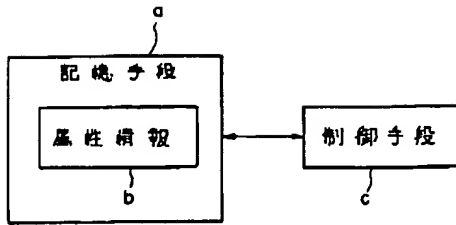
【図20】同実施例におけるレコード管理方式時の書き込みコマンド及び該コマンドに対するレスポンスデータのフォーマットを示す図。

【符号の説明】

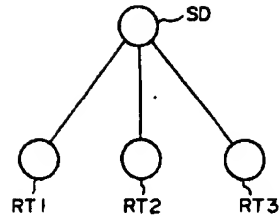
13 EEPROM（記憶手段）

14 CPU（制御手段）

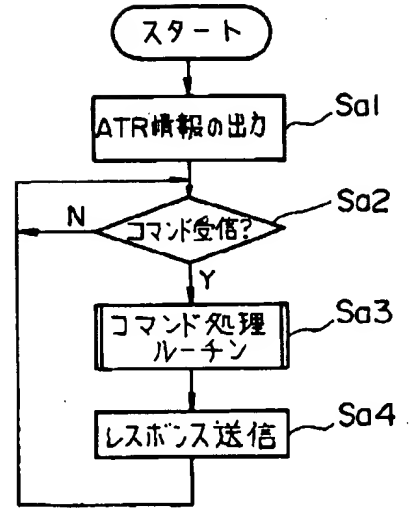
【図1】



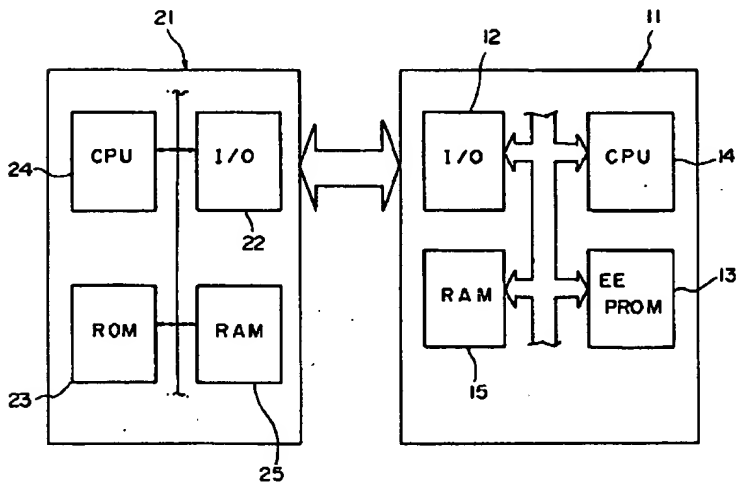
【図3】



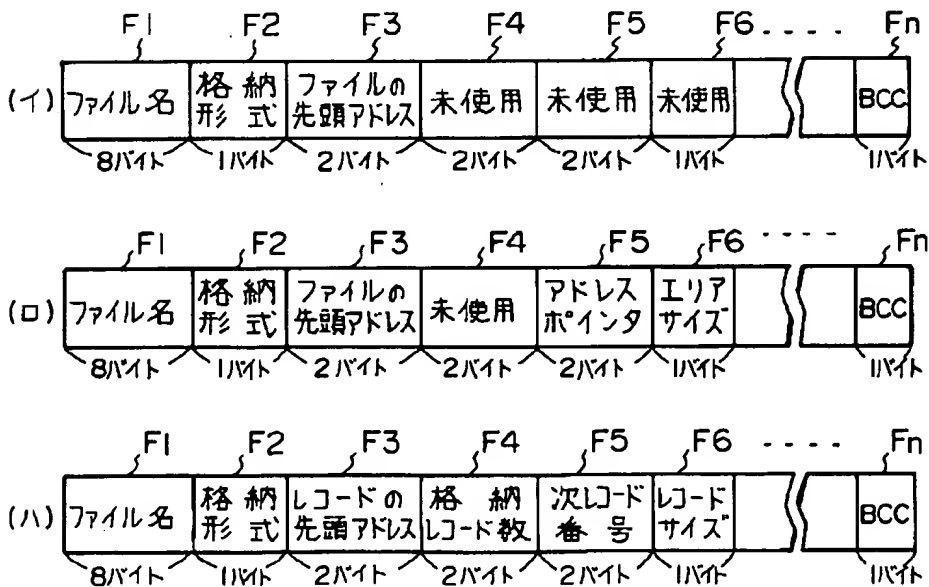
【図6】



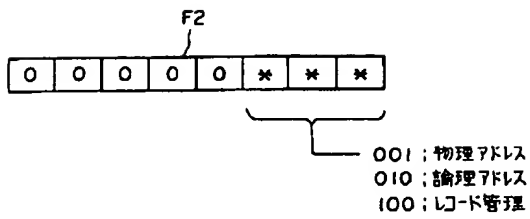
【図2】



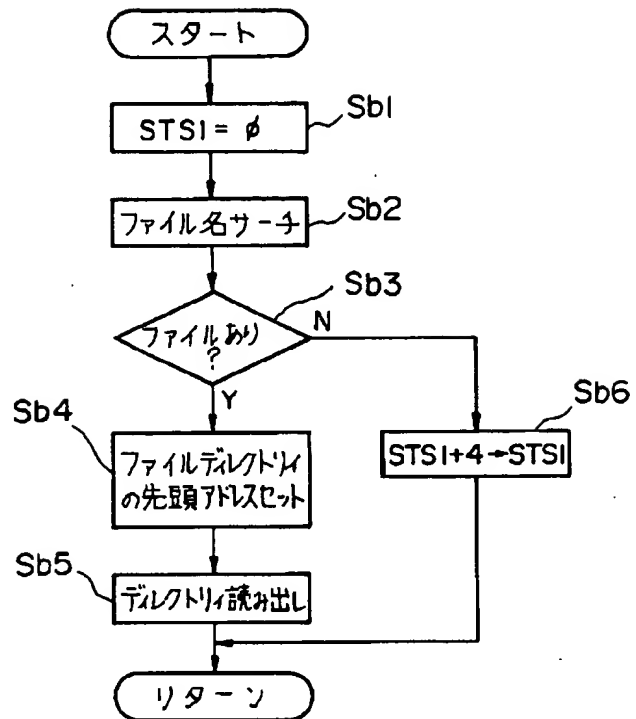
【図4】



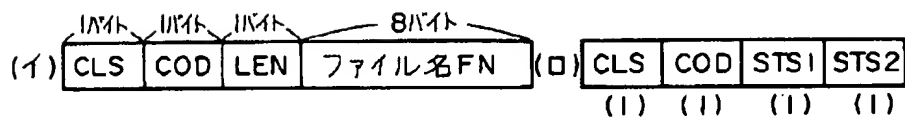
【図5】



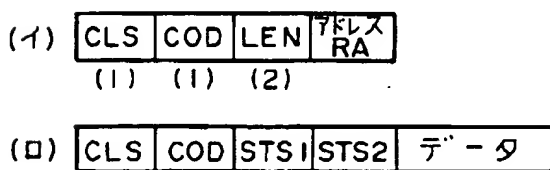
【図7】



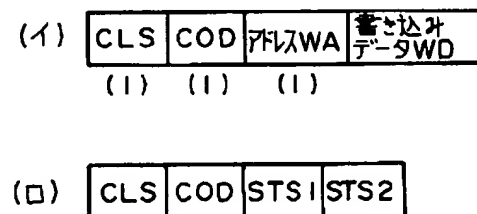
【図8】



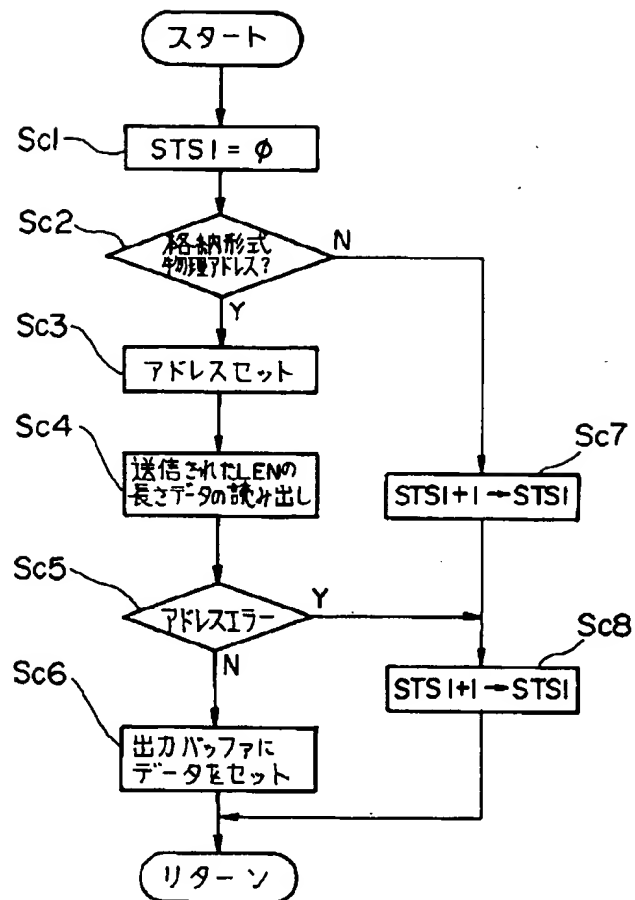
【図10】



【図12】



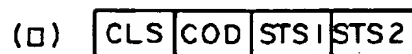
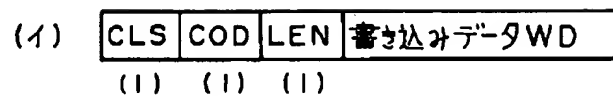
【図9】



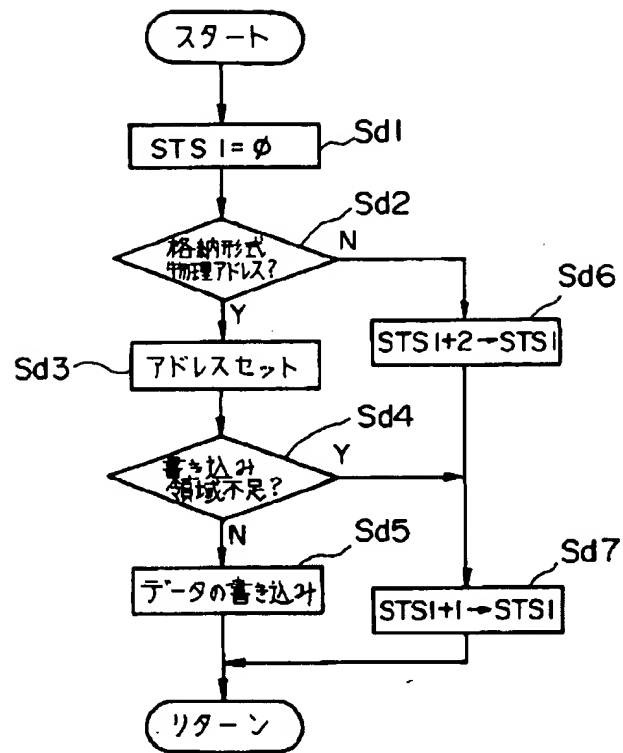
【図15】



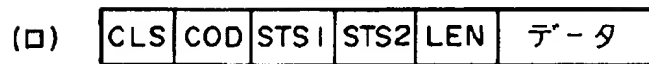
【図16】



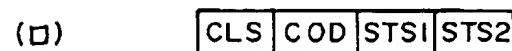
【図11】



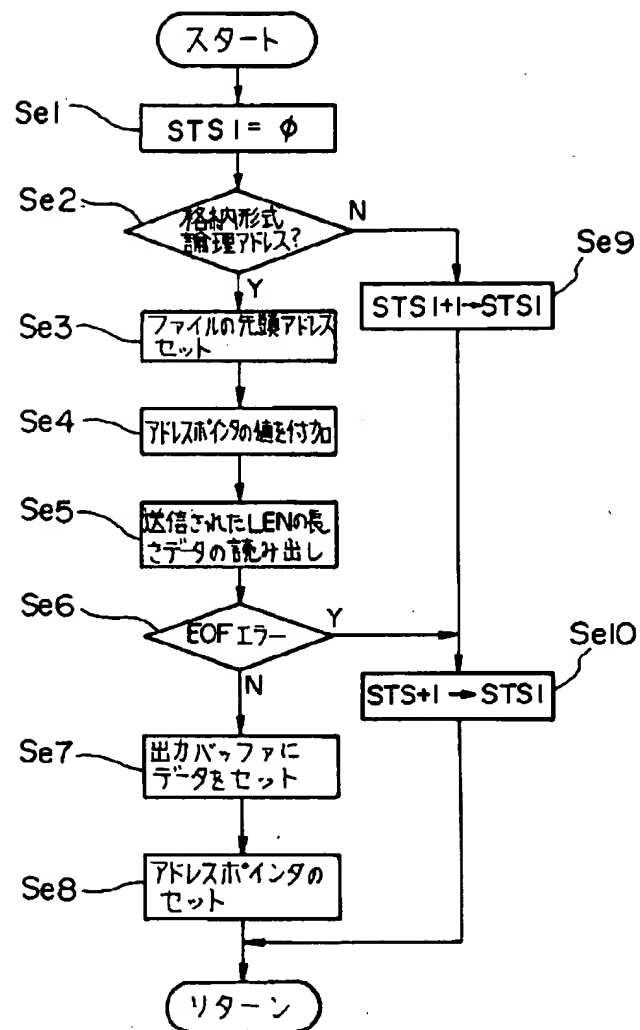
【図19】



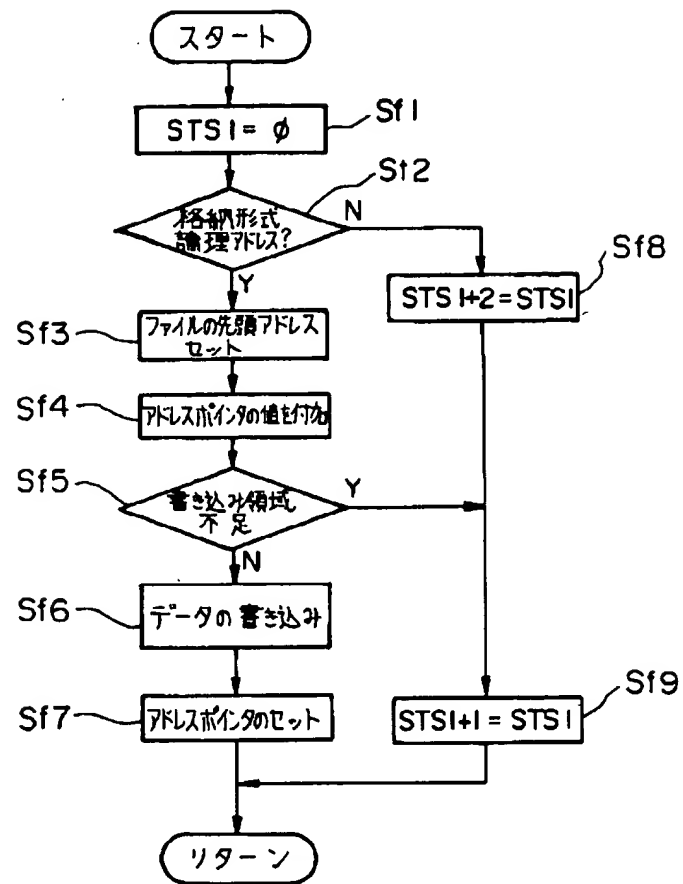
【図20】



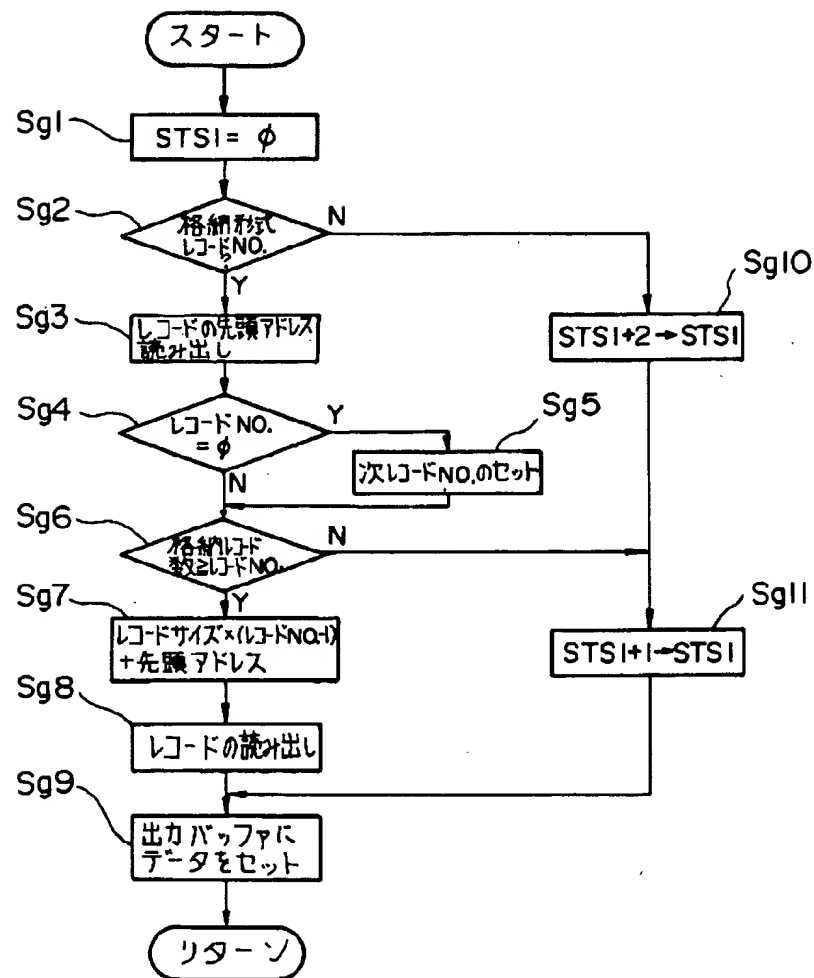
【図13】



【図14】



【図17】



【図18】

